NASA/TM—2017-219515

Assessment of the State-of-the-Art of System-Wide Safety and Assurance Technologies

Indranil Roychoudhury
Stinger Ghaffarian Technologies, Inc., Moffett Field, California

Mary S. Reveley
Glenn Research Center, Cleveland, Ohio

Nipa Phojanamongkolkij
Langley Research Center, Hampton, Virginia

Karen M. Leone
Vantage Partners, LLC, Brook Park, Ohio

June 2017

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., "quick-release" reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 757-864-6500

- Telephone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Program
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM—2017-219515

Assessment of the State-of-the-Art of System-Wide Safety and Assurance Technologies

Indranil Roychoudhury
Stinger Ghaffarian Technologies, Inc., Moffett Field, California

Mary S. Reveley
Glenn Research Center, Cleveland, Ohio

Nipa Phojanamongkolkij
Langley Research Center, Hampton, Virginia

Karen M. Leone
Vantage Partners, LLC, Brook Park, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

June 2017

Acknowledgments

The funding for this work was provided by the System-Wide Safety Assurance Technologies Project. The authors would like to thank Dr. Kamalika Das, UARC, NASA Ames Research Center, for invaluable discussions and suggestions.

This work was sponsored by the Airspace Operations and Safety Program at the NASA Glenn Research Center.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is available in electronic form at http://www.sti.nasa.gov/ and http://ntrs.nasa.gov/

Assessment of the State-of-the-Art of System-Wide Safety and Assurance Technologies

Indranil Roychoudhury
Stinger Ghaffarian Technologies, Inc.
Moffett Field, California 94035

Mary S. Reveley
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Nipa Phojanamongkolkij
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23681

Karen M. Leone
Vantage Partners, LLC
Brook Park, Ohio 44142

Abstract

Since its initiation, the System-wide Safety & Assurance Technologies (SSAT) Project has been focused on developing multidisciplinary tools and techniques that are verified and validated to ensure prevention of loss of property and life in NextGen and enable proactive risk management through predictive methods. To this end, four technical challenges have been listed to help realize the goals of SSAT, namely (*i*) assurance of flight critical systems, (*ii*) discovery of precursors to safety incidents, (*iii*) assuring safe human-systems integration, and (*iv*) prognostic algorithm design for safety assurance. The objective of this report is to provide an extensive survey of SSAT-related research accomplishments by researchers within and outside NASA to get an understanding of what the state-of-the-art is for technologies enabling each of the four technical challenges. We hope that this report will serve as a good resource for anyone interested in gaining an understanding of the SSAT technical challenges, and also be useful in the future for project planning and resource allocation for related research.

Contents

Chapter 1

Introduction

The continued growth in the passenger air travel and the air transport of cargo is dependent on improving the inherent safety attributes of current and future aircraft that will operate in the Next Generation Air Transportation System (NextGen). To address this challenge, the Aviation Safety Program (AvSP) of the National Aeronautics and Space Administration (NASA) continues to conduct foundational research and develop innovative tools, concepts, and technologies to overcome the emerging demands and issues created by the nation's transition to NextGen. As a part of the AvSP, in 2010, the System-Wide Safety and Assurance Technologies (SSAT) Project was initiated to identify the risks and provide knowledge required to safely manage increasing complexity in the design and operation of vehicles and the air transportation systems [1]. SSAT is focused on methods to assess and ensure system-wide safety of complex aviation systems. The project emphasizes proactive methods and technologies, utilizes a systems analysis approach to identify key issues, and maintains a portfolio of research leading to potential solutions. A proactive approach to managing system safety requires the ability to monitor the system continuously and to extract and fuse information from diverse data sources to identify emergent anomalous behaviors after new technologies, procedures, and training are introduced. In addition, it requires the ability to reliably predict probabilities of the occurrence of hazardous events and of their safety risks. SSAT is focused on four Technical Challenges (TCs) [1]:

- **Assurance of Flight Critical Systems**: Fill a critical gap in life-cycle development of complex systems for NextGen by developing verification and validation techniques that establish justifiable confidence that new technologies envisioned for NextGen are at least as safe as (if not safer than) the current system and provide a cost-effective basis for assurance and certification of complex civil aviation systems.

- **Discovery of Precursors to Safety Incidents**: Automated discovery of precursors to aviation safety incidents by mining massive heterogeneous data sets to enable proactive management of risk.

- **Assuring Safe Human-Systems Integration**: Enable the development of robust human-automation systems by incorporating known limitations of human performance into analysis tools. These robust systems incorporate information about human processing such as perception, attention, and cognition, as well as factors that affect human performance such as fatigue and expertise, throughout the design life-cycle of human-automation systems to increase safety and reduce validation costs in NextGen.

- **Prognostic Algorithm Design of Safety Assurance**: Exploration of a process (including algorithm design) for verifiability of system health management, thus removing obstacles to their certification, and enabling their deployment by industry to take advantage of their safety benefits.

Since its initiation in 2010, the SSAT Project has been focused on developing multidisciplinary tools and techniques that are verified and validated to ensure prevention of loss of property and life in NextGen and enable proactive risk management through predictive methods. The four technical challenges listed above will help realize the goals of SSAT.

The objective of this report is to provide a general survey of SSAT-related research accomplishments by researchers within and outside NASA to get an understanding of what the state-of-the-art is for technologies enabling assurance of flight critical systems, discovery of precursors to safety incidents, assuring safe human-systems integration, and prognostic algorithm design for safety assurance.

This report is the result of an extensive survey of literature in the research area of the four technical challenges mentioned above. While the amount of literature present in each technical area is numerous, in this report, we will focus mainly on papers that were published in the last 5 years. We believe these past 5 years have witnessed extensive research being performed in each of these four technical areas, and surveying the literature in this time-frame gives us a good idea of the state-of-the-art of research related to the SSAT-related technical challenges. We hope that this report will serve as a good resource for anyone interested in gaining an understanding of the SSAT technical challenges, and also be useful in the future for project planning and resource allocation for related research.

This report is organized as follows. Chapters 2–5 give a survey of papers in each of the four technical challenges listed above, i.e., enabling assurance of flight critical systems, discovery of precursors to safety incidents, assuring safe human-systems integration, and prognostic algorithm design for safety assurance, respectively. Chapter 6 concludes the report with an executing summary.

Chapter 2

Assurance of Flight Critical Systems

The future of both civil and military aviation brings the introduction of increasingly complex systems, the use of autonomy and autonomous systems, distributed systems, integrated systems, adaptive systems, automation, and the augmentation of human performance. NextGen operations, advanced health-management systems and advanced aircraft control techniques represent a system of greater complexity than ever developed in any domain. These complex systems have great potential to increase or decrease safety, depending on whether adequate Assurance of Flight Critical Systems (AFCS) methods and tools can be developed [2].

The goal of SSAT's AFCS Technical Challenge (TC) is to develop verification and validation (V&V) methods to assure the safety of NextGen systems in a time and cost effective manner [1]. Verification is confirmation that proposed or operational systems comply with requirements throughout the system's life cycle, while validation is confirmation that proposed system requirements, and/ or operational systems, meet the expectations of the customer and other stakeholders, accomplishing the intended purpose in the intended environment(s), throughout the system's life cycle.

The research conducted by SSAT related to AFCS has been organized under the following topics:

- Argument-based safety assurance: Provides a comprehensive, consistent approach to safety assurance across all scales from specific components to airspace operational concepts.
- Authority and autonomy: Changes to the amount of authority and autonomy especially with the introduction of NextGen will require tools that assure that operational constructs are safe and coordinated.
- Distributed systems: The increasing integration of multiple aircraft systems can produce unintended consequences. Capabilities to ensure safety-critical properties of distributed systems need to be enhanced.
- Software intensive systems: The increasing complexity in aviation systems will increase substantially with NextGen. There is a need to improve the effectiveness and lower the cost of assessing whether software meets safety objectives.

Some of the major products of this TC are as follows:

(*i*) *Static code analysis techniques for certification*: The goal is to prototype a static code analyzer that can support certification through high precision and scalability.

(*ii*) *Use of formal methods as evidence for safety cases*: The goal is to investigate how formal methods can provide evidences for safety cases. This includes establishing links between V&V results and safety goals and to relate formal method assumptions to safety cases assumptions.

(*iii*) *Formal models of organizations for analyzing human/automation roles and responsibilities*: The goal is to provide a formal framework for assessing safety in models of human/machine roles and responsibilities in organizations. The approach may involve extending existing modeling languages with formal semantics to support formal analysis.

(*iv*) *Compositional reasoning to verify the safety of software for a complete Flight Critical System*: The goal is to demonstrate that V&V of complex systems can be derived through decomposition from V&V results on system components. This will have an impact on scalability, re-use, and system evolution.

The AFCS research areas addressed in the literature survey were focused around the products listed above and include: static code analysis, distributed systems, formal methods for safety cases, human/automation formal models, compositional verification, and unified ground/air systems V&V.

2.1 Static Code Analysis

Static code analysis involves the verification of software properties for locating potentially vulnerable code without actually executing the programs. In most cases the analysis is performed on some version of the source code, and in other cases, some form of the object code, usually done by an automated tool. Static analysis does not need any test cases, does not know what the program is supposed to do, looks for violations of good programming practice, and looks for particular types of programming error.

One limitation of current static analysis tools is that they report a large number of false positives. Programmers must manually review this list and decide if they are truly bugs. Also there are often too many warnings to sort. Static analysis tools also report false negatives and harmless bugs [3]. False negatives are unreported bugs, while harmless bugs are low priority problems [4]. In addition, static analysis tools can never prove that a program is completely free of flaws, and traditional dynamic testing is still required. Static analysis should be thought of as a way of amplifying the software assurance effort. The cheapest bug to find is the one that gets found the earliest, so static analysis when used early in the development cycle will lower the cost of software assurance [5].

The NRC Decadal Survey of Civil Aeronautics identified certification for civil aviation systems as a key barrier to progress and cited the need for NASA to conduct research on methods to improve the confidence in and the timeliness of certification [6]. The standard DO-178C "Software Considerations in Airborne Systems and Equipment Certification" is the commonly used standard for software in the avionics industry. It has been elaborated by Radio Technical Commission for Aeronautics (RTCA) in a tight cooperation with European Organisation for Civil Aviation Equipment (EUROCAE) which has published the corresponding guidelines in the standard procedures instruction ED-12C. The document was released in early 2012 and replaces the standard DO-178B. DO-178C / ED-12C covers the full engineering lifecycle: planning, development requirements/design/ implementation), testing, verification and certification [7]. The DO-178C/ED-12C qualification of static analysis is a critical point of the economic equation that aeronautics industries have to solve to benefit from the technological progress in the verification domain [8].

Table 2.1 provides information on state of the art static code analysis tools. Code names, applicable programming languages, source of the code, general comments, and whether or not the code is currently being used for DO-178B or DO-178C certification. Some references are also provided.

Table 2.1. State-of-the-Art Static Code Analysis Tools

Name	Programming Language	Source	General Information	Used for DO-178B/C Certification?	Ref.
CodePeer	Ada	Adacore	An advanced static analysis tool that detects potential run-time logic errors in Ada programs.	No	[9]
CodeSonar	C/C++, Java	GrammaTech	Airbus, Boeing, NASA rely on GrammaTech CodeSonar to perform static analysis in DO-178 projects, DO-178B certification	Yes	[7]
Coverity SAVE (Static Analysis Verification Engine)	C/C++, Java, C#	Coverity, a Synopsys company	Uses abstract interpretation to identify defects in source code used on the Curiosity Mars Rover's 2 million lines of code [10]	No	[11]
ESC/Java	Java	Extended static checker for Java	Developed at Compaq Research	No	[12]
FindBugs	Java	Univ. of Maryland	Has large number of bug patterns, free software. Looks at the compiled code, the class file, and does not need the original java files.	No	[13]
Klocwork Insight	C/C++, Java, C#	Klocwork, Burlington, MA	Static analysis tool that identifies security and reliability issues.	No	[14]
Malpas	C, Ada	Atkins Global	Used in a range of industries: nuclear, airborne collision avoidance system, Astra hawk fly by wire control system, Lockheed martin C130J	No	[15]
Polyspace Bug Finder	C/C++	Mathworks	Static analysis can be used for DO-178B certification	Yes	[16]
Polyspace Code Prover	C/C++	Mathworks	Produces results without requiring program execution, code instrumentation, or test cases. Polyspace Code Prover uses static analysis and abstract interpretation based on formal methods. Can be used for DO-178B certification	Yes	[17]
Polyspace Client, Polyspace Server	Ada	Mathworks	Polyspace Client™ for Ada and Polyspace Server™ for Ada provide code verification that proves the absence of overflow, divide-by-zero, out-of-bounds array access, and certain other run-time errors in source code. They use static code analysis that does not require program execution, code instrumentation, or test cases. Polyspace products for Ada use a formal methods technique called abstract interpretation to verify code. Can be used for DO-178B certification	Yes	[18]
VectorCAST/Lint	C/C++	Vector Software	VectorCAST/Lint utilizes the powerful Lint source code analysis engine from Gimpel Software and has been extended to support the extensive list of embedded compiler environments currently integrated with the VectorCAST dynamic testing product line. Can be used for DO-178B and DO-178C certification	Yes	[19]

2.2 Formal Methods

According to RTCA DO-333, Formal Methods Supplement to DO-178C and DO-278-A, formal methods are sets of mathematically based languages and techniques for the specification, development, and verification of software aspects of digital systems [20,21]. The first work on formal methods dates back to the 1960s, when engineers needed to prove the correctness of programs. The technology has evolved steadily since then, exploiting computing power that has increased exponentially. In DO-333, a formal method is defined as "a formal model combined with a formal analysis." A model is formal if it has unambiguous, mathematically defined syntax and semantics. This allows automated and exhaustive verification of properties using formal analysis techniques, which DO-333 separates into three categories: deductive methods such as theorem proving, model checking, and abstract interpretation.

Today, formal methods are used in a wide range of application domains including hardware, railway, and aeronautics. There are an increasing number of industrial experiments on the application of formal analysis techniques to the verification of software. Model checking is beginning to be used operationally in the railway domain [22]. Certification credits for the use of formal methods in aeronautics have been obtained for Airbus A380 software [23, 24]. The use of formal methods has increased within the last five years, with barriers being usability of formal methods tools and a lack of evidence to support adoption decisions [25].

Model checking is a formal method that automatically verifies that a formal model of a system meets specifications [26]. Model checking has been used successfully in the verification of computer hardware and software applications [27], but is rarely used in human-automation verification. Limitations of model checking include complexity and formal notation expressiveness. One of the challenges facing the use of modeling checking for verification is the state explosion problem [28]. As the complexity of the system increases, the memory and time required to model the system exceeds available resources [26].

Table 2.2 provides information on state of the art model checking tools, such as the names of tools, their developers, some general information about the tools, and references that can be looked up for more details about these tools.

2.3 Human-Automation Interaction Models

Human-automation interaction (HAI) has been linked to system failure [36]. Task analytic models capture the descriptive and normative human operator behavior required to control automation [37]. These models are usually structured in a hierarchy and include models such as ConcurTaskTree (CTT) [38], operator function model (OFM) [39], or User Action Notation (UAN) [40]. Task analytic models can be used to include human behavior in formal system models along with device automation, human-device interfaces, and the operational environment. Researchers have incorporated task analytic models into formal system models of human-automation interaction systems by modeling task behavior natively in formal notation [41–43] or translating task analytic representations such as CTT [44–47] and UAN [48]. System safety properties can be verified because of the modeled normative human behavior. There are limitations to these formal verification models. Not all normative human behavior is expressed, for example, CTT and Fields' [49] task modeling notation do not support all activities and actions. Of the task modeling notations used in formal verification, only the UAN supports most relationships, but is only applicable to a limited subset of human-automation interactive systems. Techniques that rely exclusively on formal modeling notations do not have this limitation, however they require that modelers manually implement task models using notations not intended for such use.

In order to support the integration of task analyses into the formal verification of larger system models, the enhanced operator function model (EOFM) as an Extensible Markup Language-based, platform- and analysis-independent language for describing task analytic models was developed [36]. The EOFM was

Table 2.2. State-of-the-Art Model Checker Tools

Name	Developer	General Information	Ref.
SPIN	Open source software developed by Bell Labs, 1980	Spin targets the efficient verification of multi-threaded software, not the verification of hardware circuits. The tool supports a high level language to specify systems descriptions called PROMELA (short for: PROcess MEta LAnguage). Spin has been used to trace logical design errors in distributed systems design, such as operating systems, data communications protocols, switching systems, concurrent algorithms, railway signaling protocols, control software for spacecraft, nuclear power plants, etc. The tool checks the logical consistency of a specification and reports on deadlocks, race conditions, different types of incompleteness, and unwarranted assumptions about the relative speeds of processes. Awarded the ACM systems software award, in 2012	[29] [30]
PAT: Process Analysis Toolkit	National University of Singapore, 2008	PAT (Process Analysis Toolkit) is a self-contained framework for composing, simulating and reasoning of concurrent, real-time systems and other possible domains. It comes with user friendly interfaces, featured model editor, and animated simulator. Most importantly, PAT implements various model checking techniques catering to different properties such as deadlock-freeness, divergence-freeness, reachability, LTL properties with fairness assumptions, refinement checking, and probabilistic model checking.	[31]
SLAM	Microsoft Research, 2000	The SLAM project, which was started by Microsoft Research, is aimed at verifying some software safety properties using model checking techniques. It is implemented in Ocaml, and has been used to find many bugs in Windows Device Drivers. It is distributed as part of the Microsoft Windows Driver Foundation development kit as the Static Driver Verifier (SDV). SLAM uses a technique called counterexample-guided abstraction refinement, which uses progressively better models of the program under test.	[32]
BLAST	University of California, Berkley, 2002	The Berkeley Lazy Abstraction Software Verification Tool (BLAST) is a software model checking tool for C programs. The task addressed by BLAST is the need to check whether software satisfies the behavioral requirements of its associated interfaces. BLAST employs counterexample-driven automatic abstraction refinement to construct an abstract model that is then model-checked for safety properties. The abstraction is constructed on the fly, and only to the requested precision.	[33]
KRONOS	VERIMAG	KRONOS is a tool developed with the aim to assist the user to validate complex real-time systems. Real-time systems are systems that must perform a task within strict time deadlines. Embedded controllers, circuits and communication protocols are examples of such time-dependent systems. These systems are often part of complex safety-critical applications such as aircraft avionics, which are very difficult to design and analyze, but whose correct behavior must be ensured because failures may have severe consequences. Hence, real-time systems need to be rigorously modeled and specified in order to be able to formally prove their correctness with respect to the desired requirements [34].	[35]

demonstrated on a simple automobile cruise control system. Although simple, this initial step illustrated how the EOFM can be used to discover potentially dangerous problems related to human-automation interaction. The EOFM is restricted to what systems it can be applied because of the limitations of formal verification with model checking.

2.4 Compositional Reasoning

Traditional verification techniques of testing and simulation suffer from two major problems: (*i*) testing at the prototype stage where error discovery can by quite costly, and (*ii*) the inability to test for all potential interactions leaving some errors undetected until use by the end user (flight crew) [50]. Formal software verification techniques such as model checking are quite useful in detecting errors; however, increasingly complex and non-deterministic aviation systems being checked are becoming too large for these tools to verify [51]. One solution to this problem is the "divide and conquer" strategy used by compositional verification [52]. Large and complex systems are divided into smaller components (models) which are verified separately. The smaller components are then recombined and then verified by checking the assumptions of the environment. The verification of the recombined system uses assumptions about the environment, guarantees provided by the components, as well as facts provided by design patterns [53].

Research in compositional reasoning is being conducted by academia, industry, research laboratories, and government agencies [28]. The majority of compositional research seems to occur mainly within a wide range of academic institutions as well as the Research Institute for Advanced Computer Science (RIACS), and NASA. To be noted is that the RIACS is a close collaborator with NASA.

Chapter 3

Discovery of Precursors to Safety Incidents

The research goals of the Discovery of Precursors to Safety Incidents (DPSI) technical challenge is "the automated discovery of precursors to aviation safety incidents" through analyzing massive heterogeneous data sets to enable risk management [1]. The main objective of this technical challenge is to develop advanced techniques to ensure that in the future, when the complexity of the national airspace system and traffic density increases manifold, the accident rates in air transportation are still maintained at present-day (extremely low) levels. Anomaly detection from massively large mixed continuous and discrete data sets, and discovering precursors to aviation safety incidents will allow the stakeholders to proactively avoid such incidents, or be better prepared for mitigating them.

In the following, we describe the result of a literature survey to determine the state-of-the-art of data-mining based DSPSI related research and development with respect to aviation safety. Findings from the literature survey is classified into the following two topics: (*i*) anomaly detection from massive data sets, and (*ii*) discovery of precursors to aviation safety incidents. The authors request the readers refer to the actual publications referenced below for details of the different approaches that, for the sake of brevity, we summarize in this chapter.

3.1 Anomaly Detection from Massive Data Sets

The aviation system enabling the safe transportation of civilians world-wide is one of the most complex dynamic systems present today, with the number of flights, and complexity of the system slated to grow at an unprecedented level. In order to maintain current low fatality rates (about 1 per 1 million departures according to a 2006 study by Boeing), a lot of research interests is being focused on providing developing approaches to detect anomalous aviation events, so as to mitigate the adverse events and avoid loss of life and property.

Today's aviation systems generate massive sets of data. These data can be discrete, continuous, or heterogeneous, i.e., both discrete and continuous. For example, the flight data recorders (FDRs) installed on-board most modern commercial aircrafts record several hundred discrete and continuous flight parameters throughout the duration of the flight, at a sampling rate of about 1 Hz. The data recorded include information about engines, flight control systems, pilot commands, landing gear, and so on. These data sets are massive. In this section, we present data-mining approaches to anomaly detection from massive heterogeneous data sets to enable aviation safety.

Data mining massive data sets for anomaly detection has many application areas, such as, earth science [54], and aeronautics [55], among others. The basic premise of anomaly detection in data mining is to detect *outliers* using unsupervised, semi-supervised, and supervised techniques.

In unsupervised anomaly detection approaches, the data points are not labeled as 'nominal' or 'abnormal', and outlier detection can be performed using either density-based or distance-based techniques [55]. In density-based approaches, outliers are identified to be those points that lie in low density regions. A novel "inlier-based outlier detection" technique based on direct density ratio estimation is presented in [56]. Other examples of density-based outlier detection algorithms include [57–59] In distance-based approaches, 'near-by' points are identified to generate clusters, and an outlier is a point that is the farthest from all other points. Some examples of distance-based approaches are presented in [60–62].

Supervised approaches require data points to be labeled as 'nominal' or 'abnormal' so that a classifier can be trained using these data points. The ultimate goal of supervised outlier detection approaches is to use the classifier on unknown data points and correctly label them as nominal or abnormal. Neural networks [63] and decision trees [64] are examples of such classifiers. Support Vector Machines (SVMs) [65] are another example of supervised approaches, and One-class Support Vector Machines (SVMs) [66] can be used for outlier detection. Bayesian reasoning has also been leveraged to build classifiers [67]. Semi-supervised approaches require only data points that are labeled 'nominal' and do not require labeling of 'abnormal' points as well [68]. Hence, typically, semi-supervised approaches are easier to implement than the supervised ones.

In the aviation domain, it is very important to be able to detect anomalies in sequences of discrete events. One of the main algorithms that permit this is SequenceMiner [69]. SequenceMiner is an example of a supervised approach that can detect anomalous switching. Other approaches for outlier detection applied to the domain of aviation include Orca [61], Inductive Monitoring System (IMS) [70], and Multiple Kernel Anomaly Detection (MKAD) [71]. ORCA uses Euclidean distance metric to determine its nearest neighbors and any data point outside these clusters is labeled an outlier. IMS uses the Eucledian distance between monitored point and the nearest cluster (defined as a hypercube) to determine if the monitored point is an outlier or not. MKAD applies "multiple kernel learning" (MKL) [72] wherein combinations of simultaneous multiple kernels are used instead of a single kernel to improve upon the performance of 'single-kernel' SVMs.

While Orca and IMS both deal exclusively with multivariate continuous data, MKAD can be applied to both continuous and discrete data. Both IMS and Orca are distance-based, unsupervised outlier detection algorithms. Indexed-Orca (iOrca) [60] is a new method that shows at least an order of magnitude improvement in performance over that of Orca. iOrca is also applicable to heterogeneous data, just like MKAD.

Examples of Bayesian outlier detection approaches include Gaussian Process Regression [73]. GPR models the nonlinear relationships between data points and presents the prediction results as a distribution. Very recently, improvements have been made to GPR, e.g., the Block-GP algorithm [54] that allows for an order of magnitude improvement in the scalability as compared to standard GPR methods. In Block-GP, the multimodal data is partitioned into semantically meaningful partitions, and local Gaussian Processes are built off each partition. Another example of a Bayesian outlier detector example uses Hidden Semi-Markov Models (HSMM) [74], that can not only model state transitions based on observed pilot actions, but also model durations. The HSMM-based anomaly detection algorithm presented in [74] has been demonstrated to work on synthetic and flight-simulator data.

In [55], the authors applied the iOrca and MKAD algorithms to the Flight Operations Quality Assurance (FOQA) data. The FOQA database contains massive sets of data that are structured into a matrix with row corresponding to time samples sampled at 1 Hz and columns corresponding to more than 400 flight parameters that can be both continuous as well as discrete. Examples of continuous flight parameters include altitude, airspeed, roll, pitch, angle of attack, wind speed, and so on. Discrete flight parameters include landing flap position, landing gear status, etc. The two algorithms, iOrca and MKAD, work at different levels of abstractions. Hence, when applied to the FOQA data in [55], MKAD focuses on 'system-level' analysis, identifying whether or not an entire flight is anomalous. On the other hand, iOrca takes as input the time-series data from the flight recorder to identify anomalies during a particular flight.

3.2 Discovering Precursors to Aviation Safety Incidents

Once the anomaly detection is performed, interesting anomalies detected by these algorithms can be validated by domain experts. In the process of validating, analysis can be done to identify precursors of these detected anomalies. In this section, we present some approaches for discovering precursors to aviation safety incidents.

In [75], the authors present a novel inverse reinforcement learning (IRL) based approach for identifying precursors to anomalies in flight data present in the FOQA database. The IRL method uses observed behavior of an agent making decisions using a finite Markov Decision Process (MDP) [76] to determine the underlying reward function of the MDP. In order to identify precursors of anomalies, the IRL is applied on training data of experts to determine the rewards at different states, and these rewards are compared to the rewards obtained by applying IRL on the test data of non-experts. If the reward for a state in the test data is lower than that of the corresponding state in the training data, then test agent has executed a sub-optimal action. In this way, comparing the actions of the test agent to the optimal actions taken by experts, a sequence of suboptimal actions can be detected which in turn help in the discovery of precursors to adverse events. When applied to the FOQA data, this algorithm discovered precursors that were validated by experts to be correct.

The Aviation Safety Knowledge Discovery (AvSKD) Process described in [55] is designed for discovering precursors to anomalous aviation safety events, especially those related to human automation interaction. Once the two anomaly detection algorithms, MKAD and iOrca, are applied to the FOQA data, the MKAD algorithm identifies entire flights as anomalous, while the iOrca algorithm identifies the time points during a flight at which an anomaly was found. The number of anomalies detected by either algorithm can be very large depending on the input database size. A k-means clustering algorithm is applied on the large number of anomalies found, and representative examples from different clusters of anomalies are presented to the domain experts for validation. However, not all of the found representative anomalies would be interesting. Hence, once some interesting anomalies are detected, a Multivariate Time-Series Search (MTS) [77] is used to find other similar anomalies. While validating the detected anomalies of interest, the AvSKD process identifies precursors to be "combinations of conditions that increase the likelihood of potential unsafe situations in the future." [55]. The AvSKD process was successful in identifying precursors to undesirable events, such as drop in airspeed may lead to a stall warning, or a "confusion event" could have lead to a failure to reach stabilized landing criteria [55].

The Hidden Semi-Markov Model (HMSM) based anomaly approach presented in [74] develops a robust framework for detecting precursors to aviation safety incidents due to human interactions. The HMSM model weakly ordered a sequence of control actions taken by the flight crew in a flight simulator. During *normal* operations, pilot action sequences are stored in a database. Then when a new pilot action sequence is obtained, this sequence can be labeled as anomalous "if the sequence contains patterns that do not conform to the expected behavior [78]." Otherwise, the new pilot action sequence is considered to be normal.

Chapter 4

Assuring Safe Human-Systems Integration

The goal of the "Assuring Safe Human-Systems Integration" technical challenge (TC) is to "enable the development of robust human-automation systems by incorporating known limitations of human performance into analysis tools" [1]. There are four key research deliverables for this TC, which are: (*i*) methods for determining human functional state in operations; (*ii*) predictive human performance design tools; (*iii*) identification of novel human-automation interaction failures; and (*iv*) human automation design tools.

Literature from academia, industry, and other government agencies was surveyed to assess the state of the art related to this technical challenge. Over 100 publications from the time period 2010-2014 were reviewed. Over 40 of these reviewed literatures, which are refereed technical papers from technical conferences, journals, and technical reports, were mapped to the four key deliverables. The next section presents the results of the literature survey on the state-of-the-art in this technical challenge, broken down by the key deliverables.

The following sections contain the results of the literature survey by the four key deliverables. The summarized results for each deliverable are given below:

4.1 Methods for Determining Human Functional State in Operations

This research deliverable's goal is to "provide operator state indices relevant to evaluating the efficacy of flight deck technology operating concepts and evaluative scenarios. Methods will be assessed in conjunction with subjective and performance measures, including the study of flight crew physiological data during actual airline operations as part of SAA to determine indicators of operator status and performance" [1]. Literatures surveyed that may relate to this goal are presented next.

References [79] and [80] use human-in-the-loop (HITL) experiments in flight simulators for a military domain to examine operator state indices relevant to different flight deck areas. Specifically, these experiments use: (1) eye-tracking technology to measure total fixation duration on the different parts of flight deck areas; (2) the emWave-2 technology to measure physiological coherence by the heart rate variability (HRV) analysis; and (3) NASA TLX to measure subjective workload.

For a commercial civil domain, [81] uses a HTIL simulation experiment to evaluate different flight deck designs from three different commercial aircraft manufacturers (Boeing, Airbus, and Commercial Aircraft Corporation of China Ltd.) The experiment measures situation awareness using the situation awareness global assessment technique (SAGAT) and physiological index of heart rate (HR) to determine operator's states as they respond to different flight deck designs.

A NASA research team develops and validates a baseline human performance model (HPM) for pilot performance of current-day flight deck operations in NextGen environment [82–84]. The HPM, called

Man-Machine Integration Design and Analysis System (MIDAS), predicts operator's workload and visual fixations when the operator's task and procedures, the operational environment, and operator characteristics are given as its inputs.

Prediction of the changes in how pilots will interact with new flight deck automation using a neural network model is explored in [85,86]. A neural network approach is suitable for assessing complex interactions and interdependencies between different human-automation features that are not inherently obvious. The model predicts the human-automation interaction features (i.e., mental workload, task management effort, automation mode awareness, situational awareness, amount of automation crosschecking performed, and likelihood of automation-related error), when the automation, task being performed, operator characteristics, and context in which the operators are performing the automation related task are given as its input.

Another prediction tool to analyze and understand the ways pilots will interact with the new flight deck automation is CogTool [87]. CogTool predicts execution times of expert performance on frequent tasks that can be described in terms of discrete sequential actions. Based on the electroencephalography (EEG) and electrocardiography (ECG) sensors, a real-time crew cognitive workload tool called the crew workload manager (CWLM) to be displayed on the flight deck is investigated [88]. The CWLM will allow the workload of pilot flying and pilot monitoring to be visible to each other. It is an open-loop adaptive system in the sense that it acts as a detection and awareness tool so that the crew can take appropriate actions to better balance their workload.

Two competing flight deck design operational concepts, "pilot as pilot (PAP)" and "pilot as manager (PAM)", are developed in [89]. Pilot in the PAP design performs roles similar to a pilot on today's flight deck. The main concern of this design is pilot workload due to increased information processing requirements and task responsibilities for future operations. In contrast, pilot in the PAM is responsible for management of automated systems and flight deck task, with the majority of flight deck functions performed by automation. These two design concepts are to be developed in more detail so that human factors flight deck design guidelines for both concepts can be established.

There is also ongoing research looking at touch screen technology on the flight deck and seeks to quantify the reduction of flight deck workload due to the reduction of head down scan time on the flight deck display [90].

4.2 Predictive Human Performance Design Tools

The goal of this research deliverable is to "develop modeling and simulation capabilities that incorporate detailed representations of the human perceptual and cognitive performance, as well as the task, physical and procedural environment" [1]. Literatures surveyed that may relate to this goal are presented next.

[91–94] use a cognitive model (or a virtual pilot) to simulate pilot's behavior. The cognitive model helps reduce the need of expensive HITL experiments for the early concept design phase when many design candidates are evaluated and only a few candidates are down-selected for further development.

[95] uses a HITL simulation experiment to determine if eye metrics could detect fatigue before degradations in performance. Eye tracker data were analyzed using a technique known as Approximate Entropy (ApEn). "ApEn is based on the simple principle that if a time series signal can be compared with itself (heart beat data, for example) and the amount of the disorder in a comparison or change (entropy) between relative time shifts of these data is increasing, this is probably an indication of some change in the physiological state" [95]. ApEn can be used with small data samples and can be applied in real-time or on-line.

The pupillary light reflex (PLR) optical sensor for detecting or predicting human alertness and fatigue is studied in a HTIL experiment [96]. PLR is "the iris's response to changes in the intensity of light incidence on the retina" [96]. Test results provide a design guidance of a system for detecting and identifying dynamic

changes in the pupillary type reflex system. This study is a part of a mobile pupillography device intended to be assembled in cars and airplanes to detect human alertness and fatigue.

The use of functional near infrared spectroscopy (fNIRS), which measures prefrontal cortex activity, to predict mental workload was investigated in a HTIL simulation experiment [97]. The fNIRS measurement is in agreement with two well-known measures of mental workload: subjective self-report (NASA-TLX scores) and heart rate variability (HRV).

[98] provides a comprehensive review (with 231 literatures) for the state-of-the-art-data related to the correlation of different neurophysiologic variables to the mental states of car drivers or airplane pilots in the control experimental environments. Three focused neurophysiologic measurements are: the electroencephalogram (EEG), the electrooculogram (EOG), and the heart rate (HR); while the focused mental states are: mental workload, mental fatigue, and situational awareness. All reviewed literatures have mental state detection performed "offline".

As opposed to direct connection of the pilot to the measurement system for neurophysiological or physiological variables, [99] uses measurements of pilot/aircraft dynamic parameters (e.g., time series of angle or attack states, sideslip angle states, etc.) typically available in the cockpit for control and navigation purposes to detect pilot fatigue states. Rather than defining fatigue states as "rested" or "tired", fatigue can be categorized as gradual or continuous states. As such, the fatigue states are characterized by the use of fuzzy logic theory.

In [100, 101], the authors propose a system-level modeling and simulation framework to predict and model complex operations performed by teams of human and automated agents. The modeling framework starts with the concept of work domains (WMC) describing the work to be done, independent of how all agents are assigned to the work. Therefore, the framework can be used to determine the distribution of activities (function allocation) among all agents for a given concept of operations.

4.3 Identification of Novel Human-Automation Interaction Failures

This research deliverable goal is to "develop model of the cause of failure of new functionality due to HAI failure" ([1]). Literatures surveyed that may relate to this goal are presented next.

Air carrier pilots are required to periodically complete a simulator training program that focuses entirely on abnormal events that rarely happen but for which it is important to remain prepared. The training program has scripted abnormal events. The pilot's responses are appropriate and have a less pilot-to-pilot variability. However, when the abnormal events are presented in an unpredictable fashion, the pilot's responses are less appropriate and have a high pilot-to-pilot variability. [102] points out the failure of the training process for pilots to practice recognizing the abnormal events, and choosing and recalling the appropriate responses.

Motivated by loss of control in-flight fatal accidents in today's commercial aviation, [103] similarly explores the lack of training or experiences for many commercial pilots to prevent or recover from the upset situation (unusual attitudes of an aircraft). HITL centrifuge-based simulation experiments with commercial pilots participated were performed. The findings reveal that the participants' physical g cueing improves their upset recovery control. These findings can be used to design more realistic and adequate recovery trainings.

In 2013, FAA issued a requirement for pilot flying air carriers in the United States to receive training to recover from fully developed stalls by early 2019[1]. The training may occur in flight simulators. [104] compares three stall simulator models to determine if any model-dependent recovery training differences

[1] Qualification, Service, and Use of Crewmembers and Aircraft Dispatchers; Final Rule, 14 CFR Part 121, Federal Aviation Administration, Federal Register, Vol. 78, No. 218, November 12, 2013, http://www.gpo.gov/fdsys/pkg/FR-2013-11-12/pdf/2013-26845.pdf

exist. Three models vary from the conventional approach based on flight test data to the model based on combined computational aerodynamics, scaled wind tunnel data, and expert opinion that had stalled the actual aircraft. Test results show on average there is no significant difference among the models. The study recommends that if flight data is not available, the alternative model using the combined computational aerodynamics, scaled wind tunnel data, and expert opinion is feasible.

Related to NextGen, [105] explores whether the order of air traffic controller (ATC) training between the current-day procedures and the future procedures with NextGen tools affects the trainee's performance. Debriefing results of the HITL experiments reveal that future training should start with current-day procedures and delay the introduction of NextGen tools until trainees have established fundamental air traffic management skills.

Besides training as a cause of human-machine interface, [106] investigates communication failure for the Controller Pilot Data Link Communications (CPDLC) in NextGen environment. Based on an available database of 7,965 controller-pilot voice messages, two statistical models (logistic regression and hierarchical tree-based regression) were developed to predict the outcome (successful communication or miscommunication) given the twelve independent parameters describing the communication contexts (e.g., message duration, number of simultaneously open transactions, etc.)

[107] develops a control model of human-machine interface (HMI) system that incorporates both the real-time operator cognitive workload assessment and the uncertainties associated with automation reliability. The control HMI model uses Markov decision process theory to determine sequential operator actions, given different workload states and automation states, subjected to uncertainty.

Risks and benefits of advanced flight deck adaptive system were assessed in [108]. An adaptive system is defined as an automated system that has the authority to initiate changes in their own behavior. System users have a great concern that the system could behave randomly due to complex interaction rules of all system components. Three system components identified are: triggers; decision processing; and adaptations. Triggers initiate changes in the adaptation, based on the system's knowledge of context. Decision processing is an algorithmic process for system to determine when and what adaptive automation to take place. Adaptations are changes in automation behaviors and/or the human-machine interface in order to mitigate situations as identified by triggers. Trigger categories (e.g., operator initiated, operator measurement, state of the system, state of the environment, etc.) and type of adaptations (e.g., task sharing, task offloading, prioritization, etc.) are defined first. Then experts (pilots) judge risk and benefit scores for these triggers and adaptations independently. That is, the adaptation's scores are rated regardless of trigger categories. Nevertheless, the value of this research is expected to be in suggesting adaptive system issues, risks, benefits, and guidelines to be considered in designing and approving advanced adaptive systems on the flight deck.

4.4 Human Automation Design Tools

The goal of this research deliverable is to "develop more robust human-automation systems by incorporating known limitations of human performance into analysis tools" [1]. Literatures surveyed that may relate to this goal are presented next.

[109] proposes the use of ethnographic research methods to study human-machine interface. Ethnographic methods are specifically designed to study the shared norms, values, language, beliefs, habits, assumptions, and perceptions among groups of people, the cultural variables that explicitly and implicitly shape interpretations and meanings and guide behavior and relationships. The method involves site visits, interview with users, trainers, designers, and managers of the new automation technologies for commercial airline cockpits. "This research seeks to develop new models of human operators as fully embedded in socio-technical networks, leading to new ways to improve performance, training, and safety" [109].

NASA Langley Research Center's Naturalistic Flight Deck Concept works on a new man-machine interface concept, called the Haptic Flight Control System (HFCS) [110]. The HFCS uses only stick and throttle (or simply a point and shoot interface) for easily and intuitively controlling the actual flight of the aircraft without losing any of the efficiency and operational benefits of the current paradigm. The HITL experiment revealed that the HFCS does offer an attractive and viable alternative to the tactical components of today's flight management system (FMS) and autopilot control system.

[111] investigates the differences in how display clutter affects novice's and expert's visual search performance in order to improve future training and aeronautical chart design. Using existing algorithm (Color-Clustering Clutter - C3) to quantify the amount of clutter in visual displays, the HITL simulation experiment was set up and data were collected by the eye tracker devices. Results show that experts are more accurate in searching for target images but slower than novices.

Another method, called PixelAnalyzer, was developed to assess display clutter scores of Head-Up Display (HUD) [112]. Display clutter score is a composite score (by means of rank-weighted sum of ratings across dimensions) of six dimensional measures (i.e., redundancy, colorfulness, feature salience, feature dynamics, feature variability, and global density). Ranking and rating of these multidimensional measures for different HUD configurations were done by the subject pilots. NASA-TLX workload rating was used in the HITL simulation experiment. The pilot's perceived workload was correlated with the calculated composite clutter scores. Results reveal that pilot's workload and performance are lower for low and high clutter displays.

Simplification of visual clutter of aeronautical charts for arrival and departure procedures is investigated in a HITL experiment in [113]. Pilots are able to find information more quickly from the simplified chart images, where procedure-irrelevant information is excluded.

[114] analyzes three HITL datasets in order to determine influential factors in pilot's perception of display clutter on cockpit display. Operating conditions under investigation are the flight domains (fix-wing vs. vertical takeoff and landing domains) and presentation conditions (static vs. dynamic images). The analysis reveals that the perception of clutter depends on flight domains, and the dynamic presentation does not degrade the perceived clutter, but rather provided a limited mitigation of perceived clutter.

To compare multiple visual cockpit displays for a given task environment, [115] proposes a metric called Cognitive Efficiency (CE). The CE is a ratio of information reliably communicated by a display to the amount of mental resources required to process that information.

Many studies find that the benefit of cockpit automation is to relieve pilots of tedious control tasks, freeing them to focus on the big picture of the flight. Contradictorily, many studies also find that automation has led to pilot's lesser situation awareness. This is the motivation of [116] to determine how pilots make use of their free time while using automation. Results show that when higher levels of automation are used, pilot's thoughts shift from specific task-at-hand toward flight's planning ahead. However, during the successful flight performance, pilot's thoughts often drift to matters unrelated to the flight.

[117] explores the effects of cockpit weather presentation symbology on General Aviation (GA) pilot weather avoidance, weather presentation usage, and cognitive workload. Currently, there are no Federal Aviation Administration (FAA) or industry standards for the presentation of weather information in the cockpit. Any findings from this research will aid in standardization and optimization of weather presentations. Results reveal that variations in colors and weather symbology seem to affect GA pilot behavior and decision-making.

[118] summarizes guidance on human factors/pilot interface issues to be considered during the design and evaluation of avionics displays and controls for all aircraft types (14 CFR parts 23, 25, 27, and 29). These issues are categorized into ten areas. Each area represents a system of interacting elements working together to achieve a common goal. Within each area, FAA regulatory and guidance material (e.g., size, symbol, color, etc.) and other recommendations (e.g., refresh rate, update rate, etc.) are given in detail.

Table 1 provides these areas along with their description.

Table 4.1. Ten areas of human factors/pilot issues [118]

S. No.	Area	Description
1	Display hardware	Addresses topics related to the viewability, readability, and legibility of the display due to characteristics of the hardware, such as its resolution and size, or its placement and location in the flight deck.
2	Electronic display information elements and features	Addresses the design and format of information on the display, e.g. labels, symbols, and color.
3	Considerations for alerting	Provides context and examples of the use of the term "alert" in regulatory and guidance material and provides information for the design and evaluation of warnings, cautions, advisories, messages, and annunciations.
4	Organizing electronic display information elements	Describes how different displays should be arranged on the flight deck and how individual information elements and/or displays can be configured.
5	Controls	Addresses the design, layout, and operation of controls and discusses unique usability issues for specific types of controls.
6	Design philosophy	Describes prescribed human factors practices for the use of a flight deck design philosophy.
7	Intended function	Contains guidance for evaluating the intended function of a system.
8	Error management, prevention, detection, and recovery	Addresses considerations for identifying and mitigating the potential for human error.
9	Workload	Addresses minimum flight crew requirements and discusses methods for evaluating workload.
10	Automation	Discusses changes to the flight crew's role and method of operation.

Chapter 5

Prognostic Algorithm Design of Safety Assurance

Prognostics and health management (PHM) systems are important for ensuring safe and correct operation of complex engineered systems, while reducing system downtime and maintenance costs. Diagnostics and prognostics algorithms are integral components of PHM systems. Diagnsotics algorithms involve fault detection, isolation, and identification, while prognostics algorithms involve prediction of how the system will evolve in the future and estimating fault progression trajectories.

The research goals for the Prognostic Algorithm Design for Safety Assurance technical challenge (TC) is to "explore a class of new prognostic algorithms that are verifiable, thus removing obstacles to their certification and enabling their deployment by industry to take advantage of their safety benefits" [1]. The main research deliverables for this technical challenge are as follows: (*i*) establish a performance baseline for prognostic algorithms, (*ii*) investigate a systematic process for verification of prognostic algorithms, (*iii*) develop safety assurance performance metrics, and (*iv*) demonstrate how prognostic algorithms can be used for decision making.

In the remainder of the chapter, we will present the current state-of-the-art in research related to the different deliverables of this technical challenge. A large number of literature was reviewed for this exercise, and key findings from these papers are summarized below.

5.1 Performance Baseline for Prognostics Algorithms

The availability of a standardized platform for evaluating and comparing various prognostics algorithms is important for extending the state-of-the-art of prognostics algorithms in general. Being able to assess the performance of different prognostics algorithms and comparing them to baselines will also enable the verification and validation of prognostics algorithms, eventually leading towards certification of prognostic health management systems, and their deployment of these prognostics algorithms in real-world applications.

To aid this assessment, in [119], the authors developed a set of prognostics performance metrics from the point of view of practitioners applying prognostic information in health management and prognostic decision-making applications. These metrics were used in defining performance requirements for prognostics algorithms. The authors also designed and developed several testbeds for accelerated aging of components to generate benchmark run-to-failure datasets that are vital for design and testing of prognostics algorithms. One such testbed performs aging of Li-Ion batteries and measures a number of operation conditions using several sensors. This battery aging datasets are also available publicly and have been download more than 6000 times.

In [120], the authors presented the initial baseline performance results of implementing a data-driven approach to battery prognostics under random loading conditions with the hope that other researchers will build on their findings and develop other prognostics algorithms that can be compared to those presented in [120]. To the best of our knowledge, not too many other research publications report successful benchmarking of prognostics algorithms, especially on battery aging data. In [120], the authors wrote about some lessons learned, as well as pointed out the roadblocks that they encountered while developing data-driven approaches to battery prognostics. The main issue the authors report is that although some data-driven methods may seem intuitive in the beginning, when actually implemented, the results obtained may be way off from what was expected at the start. Among various approaches for implementing prognostic algorithms, data-driven algorithms are popular in the industry due to their intuitive nature and relatively fast developmental cycle. However, no matter how easy it may seem, there are several pitfalls that one must watch out for while developing a data-driven prognostic algorithm.

In [121], the authors identified that challenges of applying data-driven and model-based prognostics on different published data-sets and benchmarked these datasets in terms of how suitable these are for data-driven and model-based algorithms. The authors evaluated the same battery datasets mentioned above and concluded that while these datasets are well-suited for physics-modeling and hence model-based prognostics, developing data-driven approaches using the available datasets is difficult.

5.2 Systematic Verification Process for Prognostics

Over the past few years, a lot of research has been done on developing and analyzing prognostics algorithms. However, in order for these algorithms to be fielded in the real-world, especially in flight environments, these algorithms need to be verified and validated, and eventually certified for flight. Researchers at the Prognostics Center of Excellence at NASA Ames Research Center have developed a systematic process for verification of prognostics algorithms [122]. Prognostics algorithms that have been verified and validated using this process has eventually been deployed during flight vehicle experiments on an electric unmanned aerial vehicle (e-UAV). The goal of these experiments was to demonstrate how predictions of time-to-discharge of the e-UAV batteries could be used to maximize flight time of the e-UAV.

Requirements are crucial to the verification process. In [123], the authors presented a Systems Engineering process for developing requirements in aerospace integrated vehicle health management (IVHM) design. One of the first steps towards the verification process was to understand that specifications of requirements for prognostics algorithms are obtained through *requirements flowdown* from higher-level customer requirements in terms of performance, cost, and schedule to low-level requirements that are used by algorithm developers [124]. A simplified example of an e-UAV was used to demonstrate the requirements flowdown process in [125]. Another example of writing requirements for health management of an aircraft landing gear system as well as the verification requirements of the aircraft landing gear system was presented in [126].

Given a set of requirements, verification is the process of answering the question "are the stakeholders building the product right?" In other words, verification is a "quality control process of evaluating whether or not a product, service, or system complies with testable constraints imposed by requirements at the start of the development process" [122]. On the other hand, validation is the process of answering the question "are the stakeholders building the right product?" Hence, validation is a "quality assurance process of evaluating whether or not a product, service, or a system accomplishes its intended function when fielded in the target application domain."

In [122], the authors first distinguish between technology maturation and product development. While a prognostics algorithm is considered as a 'technology', one can only verify and validate the implementation of this technology in a 'product', such as a prognostics ad health management system (PHM system). A PHM system has many components, with the prognostics algorithms being one integral component. Then a

process is defined for verifying a prognostics algorithm as it moves up to higher maturity levels as verifying each component of a PHM system and their interactions at a particular maturity level before moving to a higher one. Any change in the components or interactions of a PHM system from one maturity level to another will require re-verification of entities that have changed.

It is observed that the verification activities need to be performed at all stages of maturity of the prognostics algorithms. However, at lower maturity levels, more effort is needed in validation of the concepts than verification, since, at these lower maturity levels, the goal to ensure that the prognostics algorithms are indeed useful in accomplishing stakeholder needs. At mid-maturity levels, more focus is on verification activities than validation as prognostics algorithms (already verified and validated at lower maturity levels) are being implemented. At higher maturity levels, relatively more effort is again on validation than verification, since by now the prognostics algorithm have been verified to be implemented correctly as per the requirements, and the focus now is to validate that the intended stakeholder needs of the prognostics algorithms have been met successfully. Therefore, the verification and validation process is an iterative process where verification and validation activities are interleaved at each maturity level, albeit at different proportions, and verification and validation of a PHM system at any maturity level assumes that it has been verified and validated at all lower maturity levels.

A systematic process for verification of prognostics algorithms is presented in [122]. While there are publications that present either only the verification of hardware [127, 128], or only software verification [129, 130], [122] focuses on both hardware and software verification. Many other researchers have pursued the verification and validation activities, however in [131, 132], it is inconsistent as to which specific activities are geared towards verification and which towards validation. Many other efforts, such as [133, 134] combine verification and validation as one task, while it is shown in [122] that this is not generally the case. There are several other researchers who use similar methods, but sometimes refer to these as verification, and sometimes as validation. The authors of [122] clearly identify the nature of distinct verification and validation activities and present a systematic process.

5.3 Safety Assurance Performance Metrics

As mentioned above, investigation of safety assurance performance metrics will allow for verification and validation of prognostics algorithms and eventually the certification of prognostics and health management system (PHM system). Towards this end, several papers, such as [119, 125, 135–137], present a number of metrics that are used towards safety assurance at different integration levels of the PHM system. The collected metrics can be classified based on their roles in verification and validation activities at different maturity levels. It was observed that at low maturity levels, a large number of performance metrics exist for performance evaluation of individual components of a PHM system (e.g., models, diagnostics algorithms, prognostics algorithms, etc.) and consequently, the verification and validation of each individual PHM system component. At higher levels of maturity, however, when the PHM system is integrated into the target system, metrics of system wide performance are not clearly defined and often depend on application scenario. In this case, the metrics for performance of the PHM system integrated into the target system will be derived from the overall system's performance metrics.

Given the understanding of the verification and validation process for PHM systems at different levels of maturity, some safety assurance metrics at each maturity level is given in Table 5.1. In the table, we adopt NASA's Technology Readiness Level (TRL) concept [138] to describe the different maturity levels. A product or technology moves up the TRL as it matures. In [138], the authors define TRL 1 through 9, where TRL 1 describes a technology at its very concept level, where TRL 9 describes the stage when the actual system is 'flight proven' through successful mission operations.

Table 5.1. Safety Assurance Performance Metrics at Different TRLs

TRL	Verification Safety Assurance Metrics	Validation Safety Assurance Metrics
1	N/A	Confirm that aging components lead to significant number of safety incidents and/or downtime; check whether prediction algorithms allow sufficient time for mitigating contingency
2	Quantify errors and confidence in computed features correlated to fault ground truth data	Identify features that correlate monotonically to measured fault growth
3	Uncertainty quantification error, modeling error, discretization error, etc.	α-λ performance prediction horizon, convergence, etc.
4	Measurement errors, manufacturing variability, channel biases, load profiles, etc.	α-λ performance prediction horizon, convergence, confidence interval, statistical hypothesis testing, reliability metric, etc.
5	Uncertainty quantification errors, modeling errors, etc.	α-λ performance prediction horizon, convergence, confidence interval, statistical hypothesis testing, reliability metric, etc.
6	Coding errors, discretization errors, sampling rate errors, communication errors, etc.	Prognostics horizon, computation time, α-λ performance robustness to system noise, prediction update rate, etc.
7	Communication errors and delays, code verification, race conditions	Number of successful flight tests, pilot's trust in prognostics decision making, etc.
8	Communication errors and delays	Number of successful safe flight tests, optimal usage of pwoer resources, quality and duration of mission, science objectives, re-planning and mitigation effectiveness, etc.
9	N/A	Effectiveness metrics, such as decrease in safety incidents, improved mission performance, cost benefits of optimal power utilization, fleet-wide maintenance schedule and cost savings, etc.

5.4 Decision Making Using Prognostics

Prognostics Decision Making involves decision making about ground operations or during flight under the presence of uncertainty given an idea of the remaining useful life. While a lot of work has been done using prognostics for decision making in domains such as financial markets [139], climate change solutions [140], and clinical medicine [141], use of prognostics for decision making in the aerospace domain is a fairly recent development, such as [142, 143].

As mentioned in [142], prognostics based decision making has been used for prognostics-enhanced control; condition-based maintenance; and automated contingency management. These applications of prognostics span several different domains, and can be considered to be the state-of-the-art at present.

5.4.1 Prognostics-Enhanced Control

In [144], the authors present a prognostics-enhanced control method where the desirability of future control outcomes is evaluated to generate control routines that optimize risk metrics derived from uncertain prognostics information about how faults grow in magnitude over time. The control algorithm was demonstrated on a skid-steered unmanned ground vehicle with winding insulation degradation due to thermal stress in the drive motors. In [145], online, real-time estimates of remaining useful life (RUL) are used to reconfigure control actions in order to trade off system performance for extended RUL. The focus of this algorithm is the

completion of critical mission objectives within a time interval determined by prognostic algorithms such that an 'acceptable level' of performance is achieved throughout the mission. In [146], the authors propose a Model Predictive Control (MPC) approach for distributing the control effort among redundant actuators. The MPC approach takes into account prognostics information about how actuators degrade to perform the redistribution.

5.4.2 Condition-based Maintenance

In [147], the authors propose an 'economic approach' for determining when to schedule maintenance actions after an anomaly is detected in the system and the remaining useful life is estimated. The proposed approach is based on Real Options (ROA) theory, where "an 'option' is a right, but not an obligation to take a particular action in the future" [147]. Real Options Analysis (ROA) is adopted from the finance domain. This approach is demonstrated using offshore wind turbines and commercial aircrafts used by commercial airlines.

Another approach for condition-based maintenance is presented in [148], where a 'post-prognostic' decision support system (DSS) allows the operator to make optimal decisions based on interactive expression of user preferences as well as the estimated prognostic state of health of a system and other variables and constraints related to system maintenance, logistics, and operations.

5.4.3 Automated Contingency Management

"Automated Contingency Management" (ACM) typically involves the capability of reconfiguration of control actions and *mission re-planning* using health state (i.e., diagnostic and prognostic) information of the system [149, 150]. In [142], the authors presents an approach to mission re-planning in the aerospace domain using methods from mathematical optimization, multidisciplinary design optimization, and game theory. Partially Observable Markov Decision Processes (POMDPs) are used to formulate the mission re-planning problem and a Probability Collectives-based technique is adopted for generating a sound policy. Prognostics information is also used for generating the action policies. While an unmanned ground vehicle was used to demonstrate this process in [142], the authors, in [143], extended this mission re-planning case study to a more challenging unmanned aerial vehicle.

Chapter 6

Summary

This report presents the results of an extensive literature survey in each of the four SSAT technical challenges, namely: (*i*) assurance of flight critical systems, (*ii*) discovery of precursors to safety incidents, (*iii*) assuring safe human-systems integration, and (*iv*) prognostic algorithm design for safety assurance. These four technical challenges have been listed to help realize the goals of SSAT, namely developing multidisciplinary tools and techniques that are verified and validated to ensure prevention of loss of property and life in NextGen, and enabling proactive risk management through predictive methods.

The objective of this report is to provide an extensive survey of SSAT-related research accomplishments by researchers within and outside NASA to get an understanding of what the state-of-the-art is for technologies enabling each of the four technical challenges. We hope that this report will serve as a good resource for anyone interested in gaining an understanding of the SSAT technical challenges, and also be useful in the future for project planning and resource allocation for related research.

In Chapter 2, we present the state-of-the-art in "Assurance of flight critical systems", such as static code analysis, formal methods, human-automation interaction models, and compositional reasoning. In Chapter 3, we survey the "Discovery of precursors to safety incidents" technical challenge and present some data mining approaches that allow for anomaly detection from massive data sets. Approaches that allow for discovering precursors to aviation safety incidents are also presented. Chapter 4 focuses on the "Assuring safe human-systems integration" technical challenge and presents some research work regarding methods for determining human functional state in operations, predictive human performance design tools, identification of novel human-automation interaction failures, and human automation design tools. Finally, in Chapter 5, we present some work related to fulfilling the research goals for the "Prognostic algorithm design of safety assurance" technical challenge, namely establishing a performance baseline for prognostic algorithms, investigation of a systematic process for verification of prognostic algorithms, development of safety assurance performance metrics, and demonstration of how prognostic algorithms can be used for decision making.

The very large number of publications in each technical challenge available in literature that were within the scope of this study, as well as the space and time constraints, made the in-depth analysis of each publication infeasible. Moreover, we are aware that we could discuss only a small selection of these papers in this report. Interested readers are requested to refer to the individual papers cited in this report for detailed understanding of each technology that we have summarized in this report.

Bibliography

1. Aviation Safety Program, "Aviation safety program: System-wide safety and assurance technologies (SSAT) project plan," 2013.

2. M. S. Reveley and K. M. Leone, "The need for research to advance the assurance of flight critical systems," NASA, Tech. Rep.

3. N. Rutar, C. B. Almazan, and J. S. Foster, "A comparison of bug finding tools for java," in *Software Reliability Engineering, 2004. ISSRE 2004. 15th International Symposium on*. IEEE, 2004, pp. 245–256.

4. S. Easterbrook, "University of Toronto, Department of Computer Science Lecture," http://www.cs.toronto.edu/~sme/csc302/notes/19-static-analysis.pdf.

5. A. Paul, "The use and limitations of static-analysis tools to improve software quality," *The Journal of Defense Software Engineering*, pp. 18–21, 2008.

6. P. Kaminski, "Decadal survey of civil aeronautics," *National Research Council, The National Academies Press, ISBN 0-309-10158-1*, 2006.

7. "Software considerations in airborne systems and equipment certification," http://www.verifysoft.com/en_DO-178C.html.

8. J.-L. Boulanger, *Static analysis of software: The abstract interpretation*. John Wiley & Sons, 2013.

9. "Adacore," www.adacore.com.

10. "Coverity: Mars Rover Curiosity's 'Space Doctors' on Bug Hunting in Space," http://www.huffingtonpost.co.uk/2012/09/27/curiositys-doctors-mars-rover-coverity_n_1919115.html.

11. "Coverity," http://www.coverity.com/products/coverity-save/.

12. "Esc java," http://kind.ucd.ie/products/opensource.ESC/Java2/.

13. "FindBugs," http://findbugs.sourceforge.net/.

14. "Klocwork insight," www.klocwork.com.

15. "Malpas," http://malpas-global.com/.

16. "Polyspace bug finder," http://www.mathworks.com/products/polyspace-bug-finder/.

17. "Polyspace code prover," http://www.mathworks.com/products/polyspace-code-prover/.

18. "Polyspace Client and Server," http://www.mathworks.com/products/polyspace-ada/.

19. "VectorCAST/Lint," http://www.vectorcast.com/resources/whitepapers/using-vectorcast-do-178b-c-software-verification.

20. Y. Moy, E. Ledinot, H. Delseny, V. Wiels, and B. Monate, "Testing or formal verification: Do-178c alternatives and industrial experience," *Software, IEEE*, vol. 30, no. 3, pp. 50–57, 2013.

21. J. M. Wing, "A specifier's introduction to formal methods," *Computer*, vol. 23, no. 9, pp. 8–22, 1990.

22. P. Behm, P. Desforges, and J.-M. Meynadier, "Météor: An industrial success in formal development," in *B98: Recent Advances in the Development and Use of the B Method*. Springer, 1998, pp. 26–26.

23. J. Souyris, V. Wiels, D. Delmas, and H. Delseny, "Formal verification of avionics software products," in *FM 2009: Formal Methods*. Springer, 2009, pp. 532–546.

24. D. Delmas and J. Souyris, "Astrée: From research to industry," in *Static Analysis*. Springer, 2007, pp. 437–451.

25. J. C. Bicarregui, J. S. Fitzgerald, P. G. Larsen, and J. Woodcock, "Industrial practice in formal methods: A review," in *FM 2009: Formal Methods*. Springer, 2009, pp. 810–813.

26. E. M. Clarke, O. Grumberg, and D. Peled, *Model checking*. MIT press, 1999.

27. E. M. Clarke and J. M. Wing, "Formal methods: State of the art and future directions," *ACM Computing Surveys (CSUR)*, vol. 28, no. 4, pp. 626–643, 1996.

28. M. S. Reveley, K. M. Leone, and S. M. Jones, "The projected impact of compositional verification on current and future aviation safety," NASA, Tech. Rep., 2013.

29. G. J. Holzmann, "The model checker spin," *IEEE Transactions on software engineering*, vol. 23, no. 5, pp. 279–295, 1997.

30. "SPIN," http://www.spinroot.com.

31. "PAT: Process Analysis Toolkit," www.patroot.com.

32. "SLAM," http://research.microsoft.com/en-us/projects/slam/.

33. "BLAST," http://mtc.epfl.ch/software-tools/blast/index-epfl.php.

34. E. A. Strunk, M. A. Aiello, and J. C. Knight, "A survey of tools for model checking and model-based development," *University of Virginia*, 2006.

35. "KRONOS," http://www-verimag.imag.fr/DIST-TOOLS/TEMPO/kronos/index-english.html.

36. M. L. Bolton, R. I. Siminiceanu, and E. J. Bass, "A systematic approach to model checking human–automation interaction using task analytic models," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 41, no. 5, pp. 961–976, 2011.

37. B. Kirwan and L. K. Ainsworth, *A guide to task analysis: the task analysis working group*. CRC press, 1992.

38. F. Paternò, C. Mancini, and S. Meniconi, "Concurtasktrees: A diagrammatic notation for specifying task models," in *Human-Computer Interaction INTERACT97*. Springer, 1997, pp. 362–369.

39. C. M. Mitchell and R. A. Miller, "A discrete control model of operator function: A methodology for information display design," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 16, no. 3, pp. 343–357, 1986.

40. H. R. Hartson, A. C. Siochi, and D. Hix, "The uan: A user-oriented representation for direct manipulation interface designs," *ACM Transactions on Information Systems (TOIS)*, vol. 8, no. 3, pp. 181–203, 1990.

41. S. Basnyat, P. Palanque, B. Schupp, and P. Wright, "Formal socio-technical barrier modelling for safety-critical interactive systems design," *Safety Science*, vol. 45, no. 5, pp. 545–565, 2007.

42. S. Basnyat, P. Palanque, R. Bernhaupt, and E. Poupart, "Formal modelling of incidents and accidents as a means for enriching training material for satellite control operations," 2008.

43. E. L. Gunter, A. Yasmeen, C. A. Gunter, and A. Nguyen, "Specifying and analyzing workflows for automated identification and data capture," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*. IEEE, 2009, pp. 1–11.

44. Y. A. Ameur, M. Baron, and P. Girard, "Formal validation of hci user tasks." in *Software Engineering Research and Practice*, 2003, pp. 732–738.

45. Y. Ait-Ameur and M. Baron, "Formal and experimental validation approaches in hci systems design based on a shared event b model," *International Journal on Software Tools for Technology Transfer*, vol. 8, no. 6, pp. 547–563, 2006.

46. F. Paternò, C. Santoro, and S. Tahmassebi, *Formal models for cooperative tasks: concepts and an application for en-route air traffic control*. Springer, 1998.

47. F. Paternò and C. Santoro, "Integrating model checking and hci tools to help designers verify user interface properties," in *Interactive Systems Design, Specification, and Verification*. Springer, 2001, pp. 135–150.

48. P. A. Palanque, R. Bastide, and V. Sengès, "Validating interactive system design through the verification of formal task and system models." in *EHCI*. Citeseer, 1995, pp. 189–212.

49. R. E. Fields, "Analysis of erroneous actions in the design of critical systems," Ph.D. dissertation, University of York, 2001.

50. M. Mach, F. Plásil, and J. Kofron, "Behavior protocol verification: Fighting state explosion," *International Journal of Computer and Information Science*, vol. 6, no. 1, pp. 22–30, 2005.

51. L. Pullum, X. Cui, E. Vassev, M. Hinchey, C. Rouff, and R. Buskens, "Verification of adaptive systems," in *Infotech@ Aerospace 2012*, 2012.

52. E. G. Cooper, B. L. DiVito, S. A. Jacklin, and P. S. Miner, "Baseline assessment and prioritization framework for ivhm integrity assurance enabling capabilities," Tech. Rep., 2009.

53. D. Cofer, "PSL for Assume-Guarantee Contracts in AADL Models," https://wiki.sei.cmu.edu/aadl/images/0/0b/RC-AADL-contractsOct2011.pdf.

54. K. Das and A. N. Srivastava, "Block-gp: Scalable gaussian process regression for multimodal data," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 791–796.

55. B. Matthews, S. Das, K. Bhaduri, K. Das, R. Martin, and N. Oza, "Discovering anomalous aviation safety events using scalable data mining algorithms," *Journal of Aerospace Information Systems*, vol. 10, no. 10, pp. 467–475, 2013.

56. S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Inlier-based outlier detection via direct density ratio estimation," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 223–232.

57. D. Ren, B. Wang, and W. Perrizo, "Rdf: A density-based outlier detection method using vertical data representation," in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE, 2004, pp. 503–506.

58. F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise." in *SDM*, vol. 6. SIAM, 2006, pp. 326–337.

59. L. Duan, L. Xu, F. Guo, J. Lee, and B. Yan, "A local-density based spatial clustering algorithm with noise," *Information Systems*, vol. 32, no. 7, pp. 978–986, 2007.

60. K. Bhaduri, B. L. Matthews, and C. R. Giannella, "Algorithms for speeding up distance-based outlier detection," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 2011, pp. 859–867.

61. S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 29–38.

62. S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM SIGMOD Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

63. G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps," *Neural Networks, IEEE Transactions on*, vol. 3, no. 5, pp. 698–713, 1992.

64. S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

65. M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

66. S. Das and N. C. Oza, "Sparse solutions for single class svms: A bi-criterion approach." in *SDM*. SIAM, 2011, pp. 816–827.

67. S. Das, K. Bhaduri, N. Oza, and A. Srivastava, "ν-Anomica: A fast support vector based novelty detection technique," in *Proceedings of the IEEE Conference of Data Mining*, 2009, pp. 101–109.

68. O. Chapelle, B. Schölkopf, A. Zien, *et al.*, *Semi-supervised learning*. MIT press Cambridge, 2006, vol. 2.

69. S. Budalakoti, A. Srivastava, and M. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 39, no. 1, pp. 101–113, 2008.

70. D. L. Iverson, "Inductive system health monitoring." in *IC-AI*, 2004, pp. 605–611.

71. S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 47–56.

72. F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 6.

73. C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," *Advances in neural information processing systems*, vol. 2, pp. 881–888, 2002.

74. I. Melnyk, P. Yadav, M. Steinbach, J. Srivastava, V. Kumar, and A. Banerjee, "Detection of precursors to aviation safety incidents due to human factors," in *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, Dec 2013, pp. 407–412.

75. V. M. Janakiraman, S. Das, B. Matthews, and N. Oza, "Identifying precursors to anomalies using inverse reinforcement learning," in *Proceedings of the 2014 Workshop on Optimization Methods for Anomaly Detection*, 2014, pp. 13–16.

76. C. C. White III and D. J. White, "Markov decision processes," *European Journal of Operational Research*, vol. 39, no. 1, pp. 1–16, 1989.

77. K. Bhaduri, Q. Zhu, N. C. Oza, and A. N. Srivastava, "Fast and flexible multivariate time series subsequence search," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 48–57.

78. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

79. W. Li, F. Chiu, Y. Kuo, and K. Wu, "The investigation of visual attention and workload by experts and novices in the cockpit," in *Proceedings of the 10th International Conference on Engineering Psychologu and Cognitive Ergonomics - Part II*, 2013.

80. W. Li, F. Chiu, and K. Wu, "The evaluation of pilots performance and mental workload by eye movement," in *Proceedings of the 30th European Association for Aviation Psychology Conference*, 2012.

81. W. Hengyang, Z. Damin, W. Xiaoru, and W. Qun, "An experimental analysis of situation awareness for cockpit display interface evaluation based on flight simulation," *Chinese Journal of Aeronautics*, vol. 26, no. 4, 2013.

82. B. F. Gore, B. L. Hooey, N. Haan, D. L. Bakowski, and E. Mahlstedt, "A methodical approach for developing valid human performance models of flight deck operations," in *Proceedings of the 2nd International Conference on Human Centered Design*, 2011.

83. B. F. Gore, "The use of behavior models for predicting complex operations," in *Proceedings of the 19th Annual Conference on Behavior Representation in Modeling and Simulation (BRiMS)*, 2010.

84. B. F. Gore, B. L. Hooey, E. Mahlstedt, and D. C. Foyle, "Evaluating nextgen closely spaced parallel operations concepts with validated human performance models: Scenario development and results, Tech. Rep. NASA TM-2013-216503, 2013.

85. K. B. Sullivan, K. M. Feigh, F. T. Durso, U. Fischer, V. L. Pop, K. Mosier, J. Blosch, and D. Morrow, "Using neural networks to assess human-automation interaction," in *IEEE/AIAA 30th Digital Avionics Systems Conference (DASC)*, 2011.

86. K. B. Sullivan, K. M. Feigh, R. I. Mappus, F. T. Durso, U. Fischer, V. L. Pop, K. L. Mosier, and D. Morrow, "Using neural networks to assess flight deck human-automation interaction," *Reliability Engineering and System Safety*, vol. 114, 2013.

87. Y. S. and L. K. A., "Predicting operator execution times using cogtool," in *17th International Symposium on Aviation Psychology*, 2013.

88. M. Dorneich, B. Passinger, C. Hamblin, C. Keinrath, J. Vasek, S. Whitlow, and M. Beekhuyzen, "The crew workload manager: An open-loop adaptive system design for next generation flight decks," in *Proceedings of the Human Factors and Ergonomics Society 55th Annual Meeting*, 2011.

89. E. Letsu-Dake, W. Rogers, M. C. Dorneich, and R. De Mers, "Innovative flight deck function allocation concepts for NextGen," in *Advances in Human Aspects of Aviation*, S. J. Landry, Ed. Boca Raton, FL: CRC Press, 2012, ch. 29.

90. S. Kaminani, "Touch screen technology in flight deck, how far is it helpful?" in *IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, 2012.

91. I. Lacko, Z. Moravek, F. Rister, J. Osterloh, F. Dehais, and S. Scannella, "Modeling approach to multi-agent system of human and machine agents: Application in design of early experiments for novel aeronautics systems," in *11th IEEE International Conference on Industrial Informatics (INDIN)*, 2013.

92. R. Hess and P. Zaal, "Visual perception and manual control," in *AIAA Modeling and Simulation Technologies Conference*, 2011.

93. S. Mamessier, K. Feigh, A. Pritchett, and D. Dickson, "Pilot mental models and loss of control," in *AIAA Guidance, Navigation, and Control Conference: AIAA SciTech*, 2014.

94. M. Lone and A. Cooke, "Review of pilot models used in aircraft flight dynamics," *Aerospace Science and Technology*, vol. 34, pp. 55–74, 2014.

95. R. A. McKinley, L. K. McIntire, R. Schmidt, A. Pinchak, J. L. Caldwell, D. W. Repperger, and M. Kane, "Evaluation of eye metrics as a detector of fatigue," Air Force Research Lab, Tech. Rep. AFRL-RH-WP-JA-2010-0002, 2010.

96. O. S. to Monitor Pupillary Light Reflex, ". rozanowski, k.; and murawski, k.:," *ACTA Physica Polonica A*, vol. 124, 2013.

97. G. Durantin, J. Gagnon, S. Tremblay, and F. Dehais, "Using near infrared spectroscopy and heart rate variability to detect mental overload," *Behavioural Brain Research*, 2014.

98. G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," *Neuroscience and Biobehavioural Reviews*, 2012.

99. M. G. Perhinschi, B. Smith, and P. Betoney, "Fuzzy logic-based detection scheme for pilot fatigue," *Aircraft Engineering and Aerospace Technology: An International Journal*, vol. 82, no. 1, 2010.

100. K. M. Feigh and A. R. Pritchett, "Modeling the work of humans and automation in complex operations," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013.

101. A. R. Pritchett, S. Y. Kim, and K. M. Feigh, "Measuring human-automation function allocation," *Journal of Cognitive Engineering and Decision Making*, 2013.

102. S. M. Casner, R. W. Geven, and K. T. Williams, "The effectiveness of airline pilot training for abnormal events," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2012.

103. W. D. Ledegang, E. L. Groen, and M. Wentink, "Pilot performance in centrifuge-based simulation of unusual attitude recovery," *Journal of Aircraft*, vol. 49, no. 4, 2012.

104. J. A. Schroeder, J. Burki-Cohen, D. A. Shikany, D. Gingras, and P. Desrochers, "An evaluation of several stall models for commercial transport training," in *AIAA SciTech, AIAA Modeling and Simulation Technologies Conference*, 2014.

105. R. C. Rorie, A. Kiken, C. Morgan, S. Billinghurst, G. Morales, K. Monk, K.-P. L. Vu, T. Strybel, and V. Battiste, "A preliminary investigation of training order for introducing nextgen tools,," *Human Interface and the Management of Information, Interacting with Information Lecture Notes in Computer Science*, vol. 6772, pp. 526–533, 2011.

106. G. Skaltsas, J. Rakas, and M. G. Karlaftis, "An analysis of controller-pilot miscommunication in the nextgen environment," in *AIAA 2011-6897, 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2011.

107. S. S. Mehta, P. E. K. Berg-Yuen, E. L. Pasiliao, and R. A. Murphey, "A control architecture for human-machine interaction in the presence of unreliable automation and operator cognitive limitations," in *AIAA 2012-4543, AIAA Guidance, Navigation, and Control Conference*, 2012.

108. M. C. Domeich, W. Rogers, S. D. Whitlow, and R. DeMers, "Analysis of the risks and benefits of flight deck adaptive systems," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2012.

109. D. A. Mindell and Z. L. Mirmalek, "An ethnographic approach to human-machine relationships in commercial aviation: Heads-up guidance and enhanced vision," in *AIAA 2011-966, 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

110. P. Schutte, K. Goodrich, and R. Williams, "Towards an improved pilot-vehicle interface for highly automated aircraft: Evaluation of the haptic flight control system," in *4th AHFE International Conference on Applied Human Factors and Ergonomic*, 2012.

111. M. R. Beck, M. Trenchard, A. V. Lamsweerde, R. R. Goldstein, and M. Lohrenz, "Searching in clutter: Visual attention strategies of expert pilots," in *Proceedings of the Human Factors and Ergonomics Society, 56th Annual Meeting*, 2012.

112. S. Kim, L. J. Prinzel, D. B. Kaber, A. L. Alexander, E. M. Stelzer, K. Kaufmann, and V. T., "Multidimensional measure of display clutter and pilot performance for advanced head-up display," vol. 82, no. 10, 2011.

113. D. C. Chandra and R. Grayhem, "Evaluation of a technique to simplify depictions of visually complex aeronautical procedures for NextGen," in *Proceedings the Human Factors and Ergonomics Society 57th Annual Meeting*, 2013.

114. D. B. Kaber, J. T. Naylor, G. Gill, C. Pankok, and S. Kim, "Influence of flight domain and cockpit display dynamics on pilot perceived clutter," *Journal of Aerospace Information Systems*, vol. 10, no. 12, 2013.

115. S. Yang, K. Shukla, and T. K. Ferris, ""cognitive efficiency in display media: A first investigation of basic signal dimensions," in *Proceedings the Human Factors and Ergonomics Society 56th Annual Meeting*, 2012.

116. S. M. Casner and J. W. Schooler, "Thoughts in flight: Automation use and pilots task-related and task-unrelated thought," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2013.

117. U. Ahlstrom and M. Dworsky, "Effects of weather presentation symbology on general aviation pilot behavior, workload, and visual scanning, Tech. Rep. DOT/FAA/TC-12/55, 2012.

118. M. Yeh, Y. J. Jo, C. Donovan, and S. Gabree, "Human factors considerations in the design and evaluation of flight deck displays and controls, Tech. Rep. DOT/FAA/TC-13/44, 2013.

119. A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for offline evaluation of prognostic performance," *International Journal of Prognostics and Health Management*, vol. 1, no. 1, p. 20, 2010.

120. A. Saxena, J. R. Celaya, I. Roychoudhury, S. Saha, B. Saha, and K. Goebel, "Designing data-driven battery prognostic approaches for variable loading profiles: Some lessons learned," in *European Conference of the Prognostics and Health Management Society*, 2012.

121. O. Eker, F. Camci, and I. Jennions, "Major challenges in prognostics: Study on benchmarking prognostics datasets," in *European Conference of the Prognostics and Health Management Society*, 2012.

122. I. Roychoudhury, A. Saxena, J. R. Celaya, and K. Goebel, "Distilling the verification process for prognostics algorithms," in *Annual Conference of the Prognostics and Health Management Society*, 2013.

123. A. Saxena, I. Roychoudhury, W. Lin, and K. Goebel, "Towards requirements in systems engineering for aerospace ivhm design," in *Proceedings of the AIAA Infotech @ Aerospace*, 2013.

124. A. Saxena, I. Roychoudhury, J. R. Celaya, S. Saha, B. Saha, and K. Goebel, "Requirements specifications for prognostics: An overview," in *Proceedings of the AIAA Infotech @ Aerospace*, 2010.

125. A. Saxena, I. Roychoudhury, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Requirement flowdown for prognostics health management," in *Proceedings of the AIAA Infotech @ Aerospace*, 2012.

126. R. Rajamani, A. Saxena, F. Kramer, M. Augustin, J. B. Schroeder, K. Goebel, G. Shao, I. Roychoudhury, and W. Lin, "Developing ivhm requirements for aerospace systems," SAE Technical Paper, Tech. Rep., 2013.

127. A. Gupta, "Formal hardware verification methods: A survey," in *Computer-Aided Verification*. Springer, 1993, pp. 5–92.

128. K. L. McMillan, "A methodology for hardware verification using compositional model checking," *Science of Computer Programming*, vol. 37, no. 1, pp. 279–309, 2000.

129. B. Bérard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, and P. Schnoebelen, *Systems and Software Verification: Model-Checking Techniques and Tools*. Springer Publishing Company, Incorporated, 2010.

130. D. R. Wallace and R. U. Fujii, "Software verification and validation: an overview," *Software, IEEE*, vol. 6, no. 3, pp. 10–17, 1989.

131. L. Tang, A. Saxena, M. E. Orchard, G. J. Kacprzynski, G. Vachtsevanos, and A. Patterson-Hine, "Simulation-based design and validation of automated contingency management for propulsion systems," in *IEEE Aerospace Conference*. IEEE, 2007, pp. 1–11.

132. M. S. Feather and L. Z. Markosian, "Towards certification of a space system application of fault detection and isolation," in *Proceedings of the 2008 International Conference on Prognostics and health management*. Citeseer, 2008, pp. 6–9.

133. R. Aguilar, C. Luu, L. M. Santi, and T. S. Sowers, "Real-time simulation for verification and validation of diagnostic and prognostic algorithms," in *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2005, pp. 1–8.

134. C. S. Byington, M. Roemer, P. Kalgren, and G. Vachtsevanos, "Verification and validation of diagnostic/prognostic algorithms," in *Machinery Failure Prevention Technology Conference*, 2005.

135. B. P. Leao, T. Yoneyama, G. C. Rocha, and K. T. Fitzgibbon, "Prognostics performance metrics and their relation to requirements, design, verification and cost-benefit," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE, 2008, pp. 1–8.

136. T. Wang and J. Lee, "On performance evaluation of prognostics algorithms," *Machinery Failure Prevention Technology*, 2009.

137. X. Guan, Y. Liu, R. Jha, A. Saxena, J. Celaya, and K. Geobel, "Comparison of two probabilistic fatigue damage assessment approaches using prognostic performance metrics," *International Journal of Prognostics and Health Management*, vol. 5, 2011.

138. J. C. Mankins, "Technology readiness levels," *White Paper, April*, vol. 6, 1995.

139. S. M. Casner and J. W. Schooler, "Thoughts in flight: Automation use and pilots task-related and task-unrelated thought," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2013.

140. ——, "Thoughts in flight: Automation use and pilots task-related and task-unrelated thought," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2013.

141. K. A. Kasmiran, A. Y. Zomaya, and Mazari, "SVM-enabled prognostic method for clinical decision making: The use of CD4 T-cell level and HIV-1 viral load for guiding treatment initiation and alteration," in *Annual Conference of the Prognostics and Health Management Society*, 2013.

142. E. Balaban and J. J. Alonso, "An approach to prognostic decision making in the aerospace domain," in *Annual Conference of the Prognostics and Health Management Society*, 2012.

143. ——, "A modeling framework for prognostic decision making and its application to uav mission planning," in *Annual Conference of the Prognostics and Health Management Society*, 2013.

144. B. Bole, L. Tang, K. Goebel, and G. Vachtsevanos, "Adaptive load-allocation for prognosis-based risk management," in *Annual Conference of the Prognostics and Health Management Society*, 2011, pp. 1–10.

145. D. W. Brown, G. Georgoulas, B. Bole, H.-L. Pei, M. Orchard, L. Tang, B. Saha, A. Saxena, K. Goebel, and G. Vachtsevanos, "Prognostics enhanced reconfigurable control of electro-mechanical actuators," in *Annual Conference of the Prognostics and Health Management Society*, 2009.

146. E. B. Pereira, R. Galvao, and T. Yoneyama, "Model predictive control using prognosis and health monitoring of actuators," in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 237–243.

147. G. Haddad, P. Sandborn, and M. Pecht, "Using real options to manage condition-based maintenance enabled by phm," in *Prognostics and Health Management (PHM), 2011 IEEE Conference on*. IEEE, 2011, pp. 1–7.

148. N. Iyer, K. Goebel, and P. Bonissone, "Framework for post-prognostic decision support," in *Aerospace Conference, 2006 IEEE*. IEEE, 2006, pp. 10–pp.

149. L. Tang, G. Kacprzynski, K. Goebel, J. Reimann, M. E. Orchard, A. Saxena, and B. Saha, "Prognostics in the control loop," in *Working Notes of 2007 AAAI Fall Symposium: AI for Prognostics*, 2007.

150. D. Edwards, M. E. Orchard, L. Tang, K. Goebel, and G. Vachtsevanos, "Impact of input uncertainty on failure prognostic algorithms: Extending the remaining useful life of nonlinear systems," DTIC Document, Tech. Rep., 2010.